Usage of Hybrid Neural Network Model MLP-ART for Navigation of Mobile Robot

Andrey Gavrilov and Sungyoung Lee

Department of Computer Engineering, Kyung Hee University, 1, Soechen-ri, Giheung-eop, Yongin-shi, Gyeonggi-do, 449-701, Korea
Andr_gavrilov@yahoo.com, sylee@oslab.khu.ac.kr

Abstract. We suggest to apply the hybrid neural network based on multi layer perceptron (MLP) and adaptive resonance theory (ART-2) for solving of navigation task of mobile robots. This approach provides semi supervised learning in unknown environment with incremental learning inherent to ART and capability of adaptation to transformation of images inherent to MLP. Proposed approach is evaluated in experiments with program model of robot.

Keywords: neural networks, mobile robot, hybrid intelligent system, adaptive resonance theory.

1 Introduction

Usage of neural networks for navigation of mobile robots is a very popular area at last time. This tendency was born in works of N.M.Amosov [1] and R.Brooks [2]. Short review of this topic may be found in [3]. This interest of using neural networks for this task is explained by that a key challenge in robotics is to provide the robots to function autonomously in unstructured, dynamic, partially observable, and uncertain environments. The problem of navigation may be divided on following tasks: map building, localization, path planning, and obstacle avoidance.

Many attempts to employ different neural networks models for solving of navigation tasks are known.

Usage of multi layer perceptrons (MLP) with error back propagation learning algorithm has some disadvantages most of them are complexity or even impossibility to relearn, slow training and orientation on supervised learning. In [4] was made the attempt to overcome some of these shortcomings by development of multi layer hybrid neural network with preprocessing with principle component analysis (PCA). This solution allows some reduce the time of learning. But rest disadvantages of MLP are remained.

In [5] A.Billard and G.Hayes suggested architecture DRAMA based on recurrent neural network with delays. This system is interesting as probably first attempt to develop universal neural network based control system for behavior in uncertain dynamic environment. However it was oriented on enough simple binary sensors detecting any events.

D.-S. Huang, L. Heutte, and M. Loog (Eds.): ICIC 2007, LNAI 4682, pp. 182–191, 2007.

We suppose that most perspective approach is usage of unsupervised learning based on adaptive resonance theory [6]. In [7] usage of this approach for building of map for navigation was proposed. The attempt of employ of model ART-2 for solving of navigation task of robot oriented on interaction by natural language was carried out [8]. But this model is dealing with primary features of images and so is sensitive to its transformations. This disadvantage leads to impossibility to use it in dynamic unknown environment for solving of such task as avoidance of obstacles using real time information from sensors. To overcome this drawback in [9] was employed multi-channel model and evaluated for solving of minefield navigation task. But in this model for every category is needed to use separate ART model. This feature limits availability of such approach, essentially in case of using visual-like sensor information.

We suggest employing of hybrid model MLP-ART2, proposed by authors and evaluated in processing of visual information [10, 11]. In this model multi-layer perceptron with error back propagation algorithm as preprocessor is used for reducing of sensitivity of ART to transformations of images from sensors. In this paper we propose usage of the model MLP-ART2 for solving of one high level task of navigation namely recognition of situation in environment with respect to position of obstacles and target and decision making about changing of direction of movement. This task is solved in combination with avoidance of obstacles solved by simple deterministic algorithms.

2 Hybrid Neural Network MLP-ART2

In our model of neural network (figure 1) the first several layers of neurons are organized as MLP. Its outputs are the inputs of model ART-2. MLP provides conversion of primary feature space to secondary feature space with lower dimension. Neural network ART-2 classifies images and uses secondary features to do it. Training of MLP by EBP (with limited small number of iterations) provides any movement of an output vector of MLP to centre of recognized cluster of ART-2 in feature space.

In this case the weight vector (center) of recognized cluster is desired output vector of MLP. It could be said that the recognized class is a context in which system try to

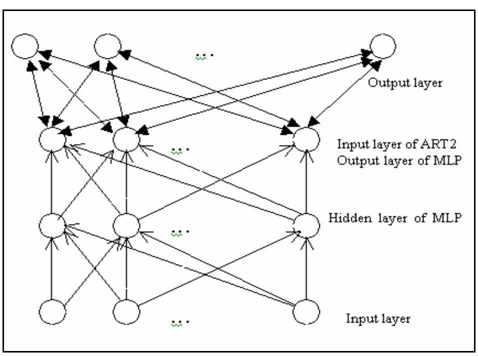

Fig. 1. Structure of hybrid neural network

recognize other images like previous, and in some limits the system “is ready to recognize” its by this manner. In other words neural network “try to keep recognized pattern inside corresponding cluster which is recognizing now”.

Action of the suggested model is described by the following unsupervised learning algorithm:

1. In MLP let the weights of connections equal to *1/n,* where n is quantity of neurons in the previous layer (number of features for first hidden layer). The quantity of output neurons N_{out} of ART-2 is considered equal zero.

2. The next example from training set is presented to inputs of MLP. Outputs of MLP are calculating.

3. If N_{out}=0, then the output neuron is formed with the weights of connections equal to values of inputs of model ART-2 (the outputs of MLP).

4. If N_{out}> 0, in ART-2 the algorithm of calculation of distances between its input vector and centers of existing clusters (the weight vectors of output neurons) is executing using Euclidian distance:

$$d_j = \sqrt{\sum_i (y_i - w_{ij})^2} \ ,$$

where: y_i – i^{th} feature of input vector of ART-2, w_{ij} – i^{th} feature of weight vector of j^{th} output neuron (the center of cluster). After that the algorithm selects the output neuron-winner with minimal distance. If the distance for the neuron-winner is more than defined a vigilance threshold or radius of cluster R, the new cluster is created as in step 3.

5. If the distance for the neuron-winner is less than R, then in model ART-2 weights of connections for the neuron-winner are updating by:

$$w_{im} = w_{im} + (y_i - w_{im}) / (1 + N_m) \ ,$$

where: N_m – a number of recognized input vectors of m^{th} cluster before. Also for MLP an updating of weights by standard error back propagation algorithm (EBP) is executing. In this case a new weight vector of output neuron-winner in model ART-2 is employed as desirable output vector for EBP, and the quantity of iterations may be small enough (e.g., there may be only one iteration).

6. The algorithm repeats from step 2 while there are learning examples in training set.

Note that in this algorithm EBP aims at absolutely another goal different from that in usual MLP-based systems. In those systems EBP reduces error-function to very small value. But in our algorithm EBP is needed only for some decreasing distance between actual and desirable output vectors of MLP. So in our case the long time learning of MLP is not required.

Algorithm EBP and forming of secondary features are executed only when image “is captured” by known cluster. So selection of value for vigilance threshold is very important. Intuitively obvious that one must be depending on transformation speed of input images and may be changed during action of system. For our architecture we used value of this parameter calculated for new cluster from distance of neuron-winner by formula

$$r = K \min d_j \ ,$$

where K is a coefficient between 1 and 2. In our experiments it was selected 1.2.

3 Simulation and Experiments

To evaluate the proposed model for selection of direction of movement with respect to position of robot, obstacles and target, experiments are conducted based on program simulation of mobile robot in 2D space for solving of navigation task, i.e. moving to target avoiding the obstacles. These experiments were provided by special program MRS developed in Delphi for simulation of mobile robots in two-dimensional simplified environment.

In our simulation following base primitives are assumed to be applied for interaction of robot with environment:

1) dist(i) – value of distance getting from i-th range sensor (one of 12 sensors);
2) *target_dist* – distance from target;
3) *target_dir* – direction to target (in degrees);
4) *robot_dir* – direction of robot's movement (in degrees);
5) *move* – command to robot "move forward in one step";
6) *turn(a)* – command to robot "turn on angle *a* (in degrees)";
7) *stop* – command to robot to halt;
8) *intersection* – situation when the target is not looked by robot directly because obstacles;
9) *target_orientation* – command to robot "turn to target direction";
10) *input* – input vector for neural network consisting of values 1 for 12 sensors, 2, 3 and 4. Length of this vector is equal 15;
11) *work_NN(input)* – start of neural network *with associative memory*, returns value of needed turn of robot in degree. The value 0 is corresponding to retain of current direction of movement, TARGET is corresponding to turn to target;
12) *ask* – prompt value of angle for rotation of robot in degree. One of possible value is SAME. It means that user agrees with value proposed by robot;
13) *current_state* – last recognized cluster or selected number of direction of movement;
14) *direction(i)* – direction corresponding to *i*th recognized cluster.

Set of distance sensors, performance of robots and obstacles are shown in fig. 4.

Algorithm of simulation of robot behavior

```
While (target_dist > 20) and not stop
  move;
  get values from sensors;
  delta = 0;
  min_distance = min(dist(0),dist(11));
  if min_distance <25 then
    if dist(0)<dist(11) then
       delta = 30
     else
```

```
            delta = -30;
        end if
     end if
  if min_distance < 5 then
        stop
    end if
    if abs(delta)=30 then
        turn(delta)
      else
        if intersection then
            Preparing vector input for NN;
            delta = work_NN(input);
            if delta = TARGET then
                target_orientation
              else
                if delta <> 0 then
                    turn(delta)
                end if
            end if
          else
            target_orientation
        end if
    end if
end while
```

End of algorithm of simulation of robot behavior

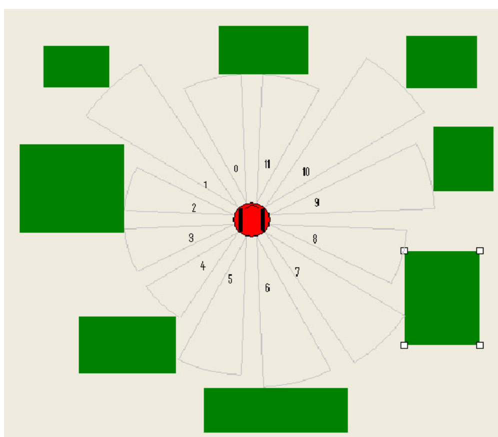

Fig. 2. Distance sensors of robot

We can see that in this algorithm two kinds of making decision are implemented – simple rules and neural network. Neural network is not used when robot may see directly target without obstacles and when robot is too closely in front of obstacle.

Otherwise, we utilized the neural network with associative memory (table) for storing of direction corresponding to cluster. In this case creating of new cluster causes the storing in this table of association between cluster (any situation) and corresponding appropriate action in this situation (selected direction of movement).

Algorithm *work_NN*

Input: input vector consisting of normalized values distance *dist(i)* from 1 for 12 sensors, *target_dist, target_dir, robot_dir.* Length of this vector is equal 15. Vigilance threshold *r*.

Output: value of angle for rotation of robot (direction).

Calculation of outputs of MLP and outputs of ART-2 (distances between input vector of ART-2 and centers of existing clusters);

```
If minimal value of outputs of ART-2 > r then
    Delta = Ask;
    r = 1.2 * minimal value of outputs of ART-2;
    If delta <> SAME then
        Creation of new cluster (with number i) with center equal input
          vector of ART-2 (output vector of MLP);
        Direction(i) = Delta;
    End if
  Else
    Delta = direction from i-th row of associative memory, where i is
       number of recognized cluster;
    Update weights of ART-2;
End if
If (minimal value of outputs of ART-2 ≤ r)
     or (delta=SAME) then
   Update weights of MLP.
   If current_state = i then
       Delta = 0;
     Else
       Delta = Direction(i); Current_state = i;
   End if
End if
```

End of algorithm *work_NN*

The experiments were conducted with two kinds of neural network – ART-2 and MLP-ART2. Respectively in first case in algorithm *work_NN* the calculation of outputs and the updating of weights for MLP are absent. The lot of experiments were conducted with different values of vigilance parameter *r* and number of iterations of EBP in MLP. Parameters of MLP are as follows: number of hidden neurons is 10, number of output neurons is 5 and the activation function is exponential sigmoid with parameter 1.

Below some screenshots of experiments are presented. The following notations are employed: 1) the trajectory of robot moving from left start point to right point which is position of target, 2) obstacle as green rectangle and 3) yellow positions of robot means that it could not select direction from associative memory itself (could not recognize known cluster) and requested prompting (supervised learning).

In fig. 3 the comparison between behavior of robot with standard model ART-2 (left) and model MLP-ART2 (right) is shown for case with one obstacle.

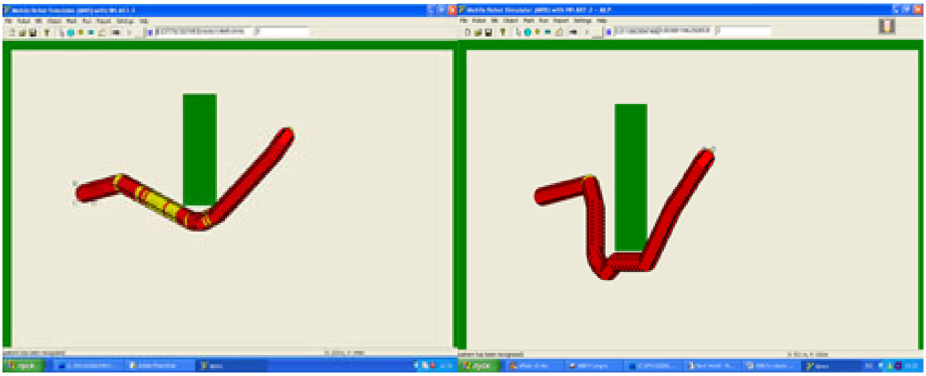

Fig. 3. The behavior of robot using ART-2 (left) and MLP-ART2 (right)

The conducting experiments show that in case of using model ART-2 without previous processing of signals from sensors the robot often asks user “what to do”. In contrast to it the model MLP-ART2 reduces number of such situations essentially after some learning and filling of associative memory by associations between created cluster and appropriate action. For configuration of environment with one obstacle and determined position of target shown in figures just 5-7 clusters are creating during learning and it is enough for practically autonomous behavior of robot independent on start position. And in this case just one iteration of EBP algorithm is enough.

Figures 4, 5, and 6 show series of screenshots obtained during sequence of experiments for case with multiple obstacles. Every experiment of series is movement of robot to target from any arbitrary point after learning during previous experiments of this series.

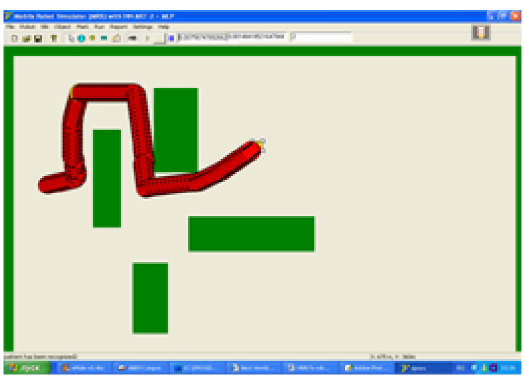

Fig. 4. The behavior of robot using the model MLP-ART2 at the first experiment of series

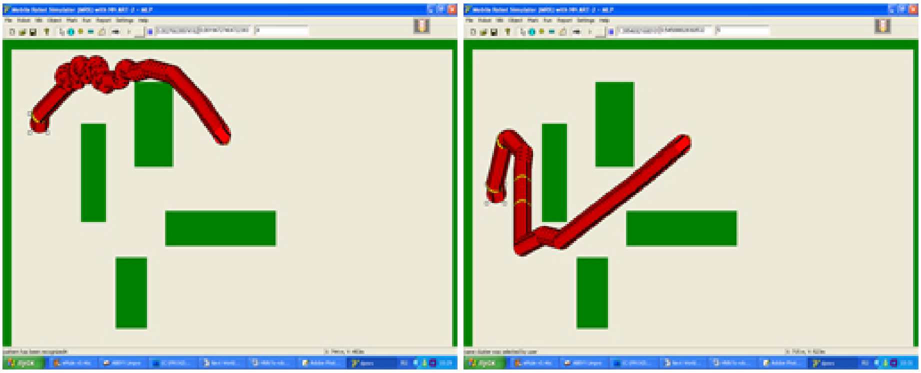

Fig. 5. The behavior of robot using the model MLP-ART2 at the 3rd and the 4th experiments of series

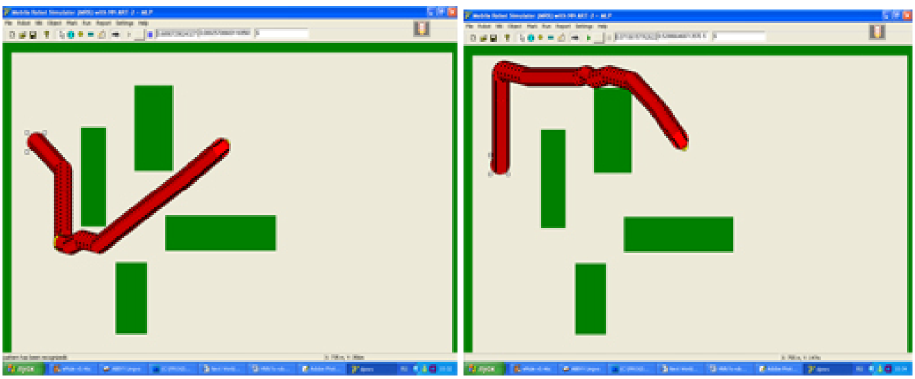

Fig. 6. The behavior of robot using the model MLP-ART2 at the 7th and the 10th experiments

Results of experiments with multiple obstacles show the decreasing of number of confusions when the robot demands assistance of operator although sometimes the robot requires help even after training before that (Figure 5, left and right). This fact may be result of not enough careful training before. Experiments show that sometimes the trajectory of movement is far from optimal essentially when environment includes many obstacles (Figure 6, right). Sometimes we can see "deadlock" when the robot can not to return from circle motion.

To overcome these disadvantages it is possible to improve logical part of control or introduce more sophisticated relations between logical rules and neural network MLP-ART2, for example, similar to proposed in [12] for hybrid expert systems. It is goal of our further researches.

4 Conclusions

In this paper we suggest and experimentally evaluate the novel approach to development of control system for navigation task of mobile robot. It is based on hybrid neural network MLP-ART2 and simple rules for navigation in specific situations. Role of MLP in this model is preprocessing of sensor signals for providing of invariant recognition of situation in environment (position of robot, obstacles and target). This architecture is further development of previous one based on ART-2 and suggested for interaction between robot and user by natural-like language for solving of navigation tasks [8]. Experiments show that using of model MLP-ART2 dramatically reduces number of situations when robot ask "what to do" although sometimes trajectory of movement is far from optimal. And just one iteration of EBP algorithm is enough for it. Probably more optimal trajectory with keeping of semi-supervised learning may be achieved by careful development of navigation rules and collaboration between ones and associative memory based on MLP-ART2. In future we plan to investigate more complex implementation of rules as knowledge based system cooperated with model MLP-ART2 through black board similar to mechanism proposed in [12]. Furthermore we plan to continue investigation the influence of parameters of our hybrid model on navigation efficiency.

Acknowledgments. This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Advancement). Dr. S.Y.Lee is the corresponding author. Thanks to PhD students Le Xuang Hung and Pho Duc Giang for their help in development of simulation program MRS for experiments.

References

1. Amosov, N. M., Kussul, E. M., Fomenko, V. D.: Transport Robot with A Neural Network Control System. In: Advance papers of the Fourth Intern. Joint Conference on Artificial intelligence 9 (1975) 1-10
2. Brooks, R.: A Robust System Layered Control System for a Mobile Robot. In: IEEE Trans. on robotics and automation RA-2 (1986) 14-23
3. Zou, A.M., Hou, Z.G., Fu, S.Y., Tan,M.: Neural Network for Mobile Robot Navigation: A Survey. In: Proceedings of International Symposium on Neural Networks ISNN-2006, LNCS 3972, Springer-Verlag, Berlin Heidelberg New York (2006) 1218-1226
4. Janglova, D.: Neural Networks in Mobile Robot Motion. In: International Journal of Advanced Robotic Systems 1(1) (2004) 15-22
5. Billard, A., Hayes, G.: DRAMA, A Connectionist Architecture for Control and Learning in Autonomous Robots. In: Adaptive Behavior 7(1) (1999) 35-63
6. Carpenter, G., A., Grossberg, S. Pattern Recognition by Self-Organizing Neural Networks, Cambridge, MA, MIT Press (1991)
7. Rui, A.: Prune-able Fuzzy ART Neural Architecture for Robot Map Learning and Navigation in Dynamic environment. In: IEEE Trans. on Neural Networks 17(5) (2006) 1235-1249
8. Gavrilov, A.V., Gubarev, V.V., Jo K.-H., Lee H.-H. In: Hybrid Neural-based Control System for Mobile Robot. In: Proceedings of 8th Korea-Russia International Symposium on Science and Technology KORUS-2004, Vol. 1, TPU, Tomsk, (2004) 31-35

9. Tan, A.H.: FALCON: A Fusion Architecture for Learning, Cognition and Navigation. In: Proceedings of IEEE International Joint Conference on Neural Networks IJCNN04, Vol. 4, 3297-3302
10. Gavrilov, A.V.: Hybrid Neural Network Based on Models Multi-Layer Perceptron and Adaptive Resonance Theory. In: Proceedings of 9th International Russian-Korean Symposium KORUS-2005, Novosibirsk (2005) 604-606
11. Gavrilov, A.V., Lee,Y.K., Lee, S.Y.: Hybrid Neural Network Model based on Multi-Layer Perceptron and Adaptive Resonance Theory. In: Proceedings of International Symposium on Neural Networks ISNN06, China, Chengdu, LNCS 3972, Shpringer-Verlag, Berlin Heidelberg New York (2006) 707-713
12. Gavrilov, A.V., Chistyakov, N.A.: An Architecture of the Toolkit for Development of Hybrid Expert Systems. In: International Conference IASTED ACIT-2005, Novosibirsk, (2005)